

Razvoj video(računalne) igre

Žic, Boris

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka / Sveučilište u Rijeci**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:195:210579>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-09-13**



Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Informatics and Digital Technologies - INFORI Repository](#)



Sveučilište u Rijeci, Fakultet informatike i digitalnih tehnologija

Sveučilišni prijediplomski studij Informatika

Boris Žic

Razvoj video igre

Završni rad

Mentor: doc.dr.sc. Miran Pobar

Rijeka, 17.09.2023.

SADRŽAJ

Zadatak završnog rada	3
Sažetak.....	4
Ključne riječi.....	4
Uvod.....	4
Proces i faze razvoja igre.....	5
Ideja.....	7
Ideja projekta	8
Program za razvoj.....	8
Dokument dizajna igre (GDD).....	9
Popis potrebnih dokumenta za razvoj.....	10
Datoteke za razvoj.....	10
Datoteke slika.....	10
Datoteke muzike.....	12
Programiranje.....	13
Testiranje i promjene.....	18
Prodaja i distribucija.....	19
Rasprava (Motivacija i poruka programerima).....	19
Zaključak	20
Opis oznaka.....	26
Popis slika	27
Programi	29
Izvori	29
Prilozi	29



Sveučilište u Rijeci
**Fakultet informatike
i digitalnih tehnologija**
www.inf.uniri.hr

Rijeka, 24.5.2023.

Zadatak za završni rad

Pristupnik: Boris Žic

Naziv završnog rada: Razvoj računalne igre

Naziv završnog rada na engleskom jeziku: Development of a computer game

Sadržaj zadatka: Proučiti faze razvoja računalnih igara i alate za njihov razvoj. Osmisliti svoju računalnu igru, odabrati odgovarajući alat za njen razvoj te ju izraditi. Opisati razvijenu igru, dizajn igre, mehanike i način napredovanja kroz igru.

Opisati postupak razvoja igre, s naglaskom na važne dijelove funkcionalnosti i vizualnog izgleda, poput kretanja likova, grafičkih efekata i sličnog.

Mentor

Doc. dr. sc. Miran Pobar

Voditelj za završne radove

Doc. dr. sc. Miran Pobar

Zadatak preuzet: 24.5.2023.

(potpis pristupnika)

Sažetak

U ovome radu opisan je detaljan postupak za izradu i razvoj video igre. Biti će opisane ideje igra i promjene. Opisan je odabir stvari poput Game engina, potrošnje budžeta, odabir programa i način izrade. Detaljno je opisano korištenje nekih programa koji su odabrani. Prikazan je način rada sa blueprintovima i nekim asetima.

Ključne riječi

2D, Animacije, Aria of the Soul, Blueprint, BoscaCoil, Game engine, Indie developer, Paper2D, PaperZD, Photoshop, Sidescroller, Unreal Engine5.

Uvod

Motivacija za ovaj zadatak je izrada video igre i prikaz cijelog razvoja video igre sa sjedišta Indie developera (Osoba ili mali team koji razvija igru s minimalnom vanjskom suradnjom te nisu dio velike, poznate, kompanije).

Zadatak je uzet kako bi bila prikazana izrada video igre i dokazano da ta izrada u modernom dobu može biti vrlo lagana. Ali za napraviti vlastitu ideju i pravu video igru, koja je nešto više nego običan projekt, mnogo je teže nego započeti biti developer video igra.

Izrada video igre ili Game development je proces iz kojeg nastaje video igra. Taj proces odrađuju developeri, bilo to jedna samostalna osoba ili cijeli tim zaposlenika. Taj proces traje između par sati i par godina ovisno o veličini i kompleksnosti ideje koja se pokušava pretvoriti u video igru, no taj proces počinje i prije nego što se otvore programi za izradu. Proces počinje još u razgovoru, sa prijateljima u kafiću, sa kolegama na sastanku, iznenadno u mislima osobe, taj proces počinje kada se prvi izgled ideje stvori u nečijim mislima. Tada ta ideja mora postati nešto na čemu se može raditi, ideja se razvija kroz razgovore u pravu cijelu ideju dok ne postane kompletirana i dok se netko ne odluči da želi preći u izradu. Kada ta ideja pređe sa misli na papir, tada počinje proces zvan game development. Unutar tog procesa ima mnogo problema koje programeri, artisti, muzičari, i mnogi drugi pokušavaju riješiti kako bi izradili tu igru. Više o tim procesima će biti objašnjeno unutar ovog zadatka.

Za ovaj zadatak pokušat će se prikazati problem koji stoji na putu bilo koje osobe koja želi proizvesti svoju igru i adresirati probleme s kojima se bore programeri igara.

Svrha ovog projekta je pokazati težinu izrade igre svima koji žele se upustiti u izradu video igre, prikazati koliko puno problema stoji na njihovom putu, izkazati koliko je teško biti programer video igra. Dok je cilj ovog projekta upravo suprotan, prikazati koliko je lagano biti programer video igra i dati detaljan putokaz za pretvoriti svoju prvu ideju u video igru.

Primarni fokus je prikazati kako izraditi igru. Za taj argument odlučio sam napraviti svoju igru, kako bih mogao to napraviti iz perspektive osobe koja i pokušava izraditi igru.

U nastavku sljedi izrada igre tipa sidescroller2D unutar UnrealEngine5, a to će se odvijati kroz razradu teme sljedom: Proces i faze razvoja, generalne ideje, ideja ovog zadatka, odabir programa za programiranje igre, GDD, popis aseta, opis stvaranja aseta, opće testiranje igra i opis prodaje i održavanja igre.

Proces i faze razvoja igre

Razvoj video igra je dugačak proces pun svakakvih problema zato se odvaja u faze razvoja. Te faze su: pred-produkcija ili dizajniranje, pitch ili predstavljanje ideje, koncept ili plan za razvoj, dokument razvoja koji je pismeni format ideje, prototip za prikazat ideju kao igru, produkcija, dizajniranje po prototipu, programiranje, izrada izgleda scenarija, stvaranje slika i muzike, testiranje, prvi igriv dio, popravak problema u kodu, drugo izdanje, završavanje i usavršavanje, izbacivanje završnog produkta i održavanje.

1. Faza pred-produkcije ili dizajniranja je planska faza koja označava početak razvoja igre i koja se fokusira na ideju i odgovor na pitanje, "O čemu se igra radi?". U ovoj fazi tim se fokusira na kreativnu stranu projekta i u ovoj fazi se još izrađuje i jednostavna dokumentacija koja opisuje zadatke, rasporede i procjene vremena za tim programera. To zovemo produkcijski plan. Ova faza se dešava prije dobivanja budžeta, ali neki izdavači zahtijevaju produkcijske planove unutar ove faze.
2. Faza predstavljanje ideje ili pitch je kratki sažetak ideje koji prezentira mogućnost monetizacije igre i detalje zašto je igra profitabilna. Ova faza može biti verbalna ili pismena, pitch ne zahtijeva cijeli dokument već samo dobro predstavljanje ideje i njenih razloga. Pitch se često dešava prije nego što se službeno krene u razvoj. To je proces u kojem se developeri igre pokušavaju predstaviti potencijalnim investitorima ili izdavačima. Za to nije potrebna dokumentacija, ali dobro strukturiran pitch igra veliku ulogu u privlačenju financija i partnerstva.
3. Faza plana za razvoj ili koncept je detaljniji dokument od pitch-a i ima informacije o većini igre poput: tipa igre, opis igre, priče igre i lokacije, za koga je priča, na kojoj platformi je izrađujemo te što nam treba za izradu. Ovo je prvi pravi dokument prije GDD koji se izrađuje za igru. Unutar ove faze programeri već započinju izradu nekih prototipa.
4. Faza pisanja dokumenta razvoja ili (dizajn dokument igre)GDD koji opisuje koncept igre i veće elemente. Nekad GDD posjeduje i skice igre koje vizualiziraju različite aspekte igre. U GDD se nalazi sve iz faze koncepta i pitch, mehanike igre, karakteristike likova, efekti, umjetnost, glazba, ova faza je skup svih informacija vezanih za igru i potrebnih za razvoj igre.
5. Faza prototip je najčešće napravljena za testiranje ideja i koristi se kao dokaz koncepta. Npr. koristi se za testiranje ako fizika kretanja radi kako je zamišljena, ili za provjeru ako akcije glavnog lika su moguće izvesti na zamišljen način ili se moraju promijeniti. Prototipovi se često izrađuju kako bi se testirali mehanizmi igre, i omogućuje programerima da brzo identificiraju potencijalne probleme i prilagode koncept prema potrebama.
6. Faza produkcije je glavna faza razvoja video igre. U toj fazi se proizvode asseti i programiraju glavni dijelovi igre. Tim započinje izradu svih elemenata igre, uključujući modele, teksture, animacije, programiranje gameplaya i ostale komponente. Ovdje se stvaraju svi asseti koji će činiti igru.
7. Faza dizajniranja po prototipu je nastavak produkcije sa dodatcima testiranih koncepta iz prototipa. Ako su se tijekom testiranja identificirali problemi ili potrebne promjene, sada se primjenjuju kako bi se poboljšala igra. To može uključivati izmjene u dizajnu levela, gameplay mehanikama ili vizualnom stilu.

8. Faza programiranja je trajna faza koje traje cijeli projekt. Ovdje se stvaraju prototipi, dodaju novi koncepti i programiraju svi dijelovi igre, od koncepta do grafike. Programeri rade na tome da se sve komponente igre ponašaju kako je zamišljeno.
 9. Faza izrada scenarija usmjerena je na stvaranje razina i sadržaja koje igrači vide tijekom igre. Prvi level često traje najduže, jer u njemu se razvijaju svi koncepti, koji se kasnije mogu samo primjeniti u druge levele. Izrada scenarija zahtijeva više nego samo slaganje kako izgleda level, tu se uključuje sve što ima funkciju u levelu, od izgleda, funkcije, pa do elemenata s kojima likovi surađuju.
 10. Faza stvaranje slika se odvija dva puta tijekom produkcije. Prvo je stvaranje konceptnih slika koje pomažu definirati izgled igre. Nakon toga je stvaranje završnih slika od konceptnih koje će se koristiti u konačnom projektu.
 11. Faza stvaranje muzike je faza koja obuhvaća stvaranje različitih audio elemenata igre, odvojeno u tri kategorije. Sound efekti koji repliciraju zvukove koji se dešavaju u interakcijama kao hodanje. Muzika koja svira cijelo vrijeme u pozadini. I voice-over koji davaju dojam stvarnih likova i njihovog govora.
 12. Faza testiranje je kontinuirani process koji se provodi tijekom programiranja i nakon programiranja. Testiranje je važno kako nebi bilo problema sa igrom nakon što programiranje završi. Tester i igraju igru kako bi identificirali bugove, probleme u mehanikama ili neusklađenosti dizajna.
 13. Faza prvog igrivog dijela ili alpha je prva završna iteracija igre koja se izrađuje i testira. Alpha verziju često testiraju privatni tester i koji pomažu identificirati preostale probleme.
 14. Faza popravka koda se izvodi nakon što se testira alpha verzija igre. Programeri rade na ispravcima koda i rješavanju problema koji su se pojavili tijekom testiranja. Tu se popravljaju većina koda i problema koji se nađu tokom alpha verzije.
 15. Faza drugog izdanja ili beta je druga završna iteracija igre koja se izgradi i testira. U testiranje beta verzije uključeni su i vanjski suradnici, game tester i.
 16. Faza završavanje je programiranje cijele igre. Ona obuhvaća finalno programiranje i ispravke kako bi se osiguralo da igra bude spremna za prodaju.
 17. Faza usavršavanje se događa ako postoji ostatak vremena između faze završavanja i faze prodaje. Ova faza omogućuje dodavanje dodatnih značajki i poboljšanja kako bi se igra usavršila prema originalnom konceptu.
 18. Faza prodaje i održavanje je stavljanje završene igre na jednu od platforma za prodaju igre i na tržište. Nakon lansiranja, tim developera nastavlja dodavati nadogradnje i ispravke kako bi održao kvalitetu igre i pratio povratne informacije igrača.
- Sve ove faze čine cjelokupan proces razvoja video igre, a svaka od njih ima svoju važnu ulogu u stvaranju kvalitetnog i uspješnog igraćeg iskustva.
- Za toliki proces postoje posebni ljudi u timu koji se sastoji od: producenta, izdavača, developera, dizajnera, umjetnika, programera, dizajnera levela, muzičara i testera. Svaka osoba u timu ima svoj posao koji moraju napraviti kako bi tim uspijeo. Naravno svaka osoba ne radi sama na svojem dijelu već se mora dobro posavjetovati s ostatkom tima.

U nastavku je prikazano kako je to odraditi sam, što znači da će se sve odvijati jedno nakon drugog, a ne kako se odvija u timu koj radi zajedno na više segmenata odjednom.

Ideja

Kako bih započeli razvoj video igara moramo imati ideju što želimo napraviti. Početna ideja nije ništa više nego rečenica, "Ne bi li bilo zabavno, uzbudljivo ili interesantno pretvoriti neku ideju u video igru?". Ali kao i sa svakom drugom idejom, zato što se može napraviti, ne znači da se treba. Dobri primjeri su današnje igre koje su izrađene zbog novaca, popularnosti ili samo zato jer se moglo. Igre koje imaju previše nastavka, svi današnji battle royal(Igra sa mnogo osoba koje se bore za prvu poziciju sa zatvarajućim krugom koji gura igrače u sredinu mape) ili RPG(Role Playing game, igra u kojoj svaki sudionik uzima ulogu lika koji može utjecati na imaginarni svijet igre) igre poput Assassin's Creed Valhalla(RPG igra od developera Ubisoft, dvanaesta instalacija u Assassin's Creed seriji igara) i prošlih 2 nastavka izrađene su zbog popularnosti žanra i novaca.

Ova slika prikazuje uspjeh igara koje su izrađene zbog popularnosti.



ASSASSIN'S CREED VALHALLA PlayStation 5

Ubisoft | Release Date: Nov 12, 2020 | Also On: PC, PlayStation 4, Stadia, Xbox One, Xbox Series X

Summary	Critic Reviews	User Reviews	Details & Credits	Trailers & Videos
	84 Metascore Generally favorable reviews based on 8 Critic Reviews What's this?	7.7 User Score Generally favorable reviews based on 2096 Ratings Your Score: 0		

Slika 1.: prikaz rezultata igre Valhalla

Te igre su izrađene zbog krivih razloga, neke uspiju, no većina ih je samo prosječna kako bi se uklopili. Te igre poput LOTR Gollum(Akcijsko-avanturistička igra od developera Daedalic, neuspješna igra zbog koje ti developer više ne proizvode igre), za koju je malo reći da je loša igra, spadaju u kategoriju igre koja se napravila samo zato što se mogla iako se i nije trebala napraviti.

Ova slika prikazuje neuspjeh igra koje su izrađene iz krivih razloga sa malo osjećaja.



THE LORD OF THE RINGS - GOLLUM PC

Daedalic Entertainment, Nacon | Release Date: May 25, 2023
Also On: PlayStation 4, PlayStation 5, Xbox One, Xbox Series X

Summary	Critic Reviews	User Reviews	Details & Credits	Trailers & Videos
	38 Metascore Generally unfavorable reviews based on 37 Critic Reviews What's this?	1.8 User Score Overwhelming dislike based on 169 Ratings Your Score: 0		

Slika 2.: prikaz rezultata igre Gollum

Ali i loše igre su nekada zanimljive, ako su u njih uloženi trud i srce. S tim razlogom ova faza je najduža faza razvoja videoigara. Iskreno rečeno, ova faza nikada ne završi. Ona se prostrinje kroz cijeli projekt i mjenja njegov izgled u svakom trenu.

U fazi ideje razmišljamo što želimo pretvoriti u videoigru, bilo to film(Shaolin vs wutang 2*), knjiga(Witcher*), osobni osjećaji(Binding of Isaac*), druga igra(pubg*-fortnight*), ili kompletno nova idea(Minecraft*). Nakon što odaberemo ideju (U ovom slučaju nova idea o osobnim osjećajima Aria of the Soul*) moramo tu ideju razvit. O čemu se ta idea radi i kako ćemo tu ideju prikazat.

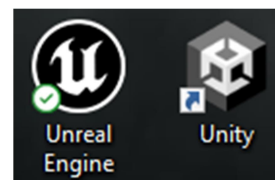
Ideja projekta

Za ovu ideju odabrali smo pričanje priče kroz akcije glavnog lika, a za žanr ove video igre odabrali smo 2D(Dvodimenzionalni pogled na svijet) sidescroller(Tip igre u kojem se svijet pomiče lijevo i desno po ekranu, najčešće viđen u 2D igrama). Ideja igre je bila napraviti lika koji se ne sijeća svoje prošlosti i depresivan je. Zbog njegove depresije naš glavni lik ima problema sa progresiranjem, zbog toga ako je ostavljen na nekoliko sekundi, on odustane od kretanja i sjedne. Isto kako i sjedne, našeg glavnog lika je teško dignut na noge, te mora se čekati da ustane. Priča se nastavlja sa time da je cijelo selo depresivno. U selu su kovač, koji je izgubio najdraži čekić, i klesar, koji nemože raditi jer nema alat za raditi. Glavni lik uz pomoć igrača prolazi kroz selo i primjećuje stanje sela. Kako bi glavni lik nastavio dalje mora proći rijeku na kraju sela, nažalost most je urušen i jedina osoba koja ga može popraviti je depresivna jer nema alat. Glavni lik pomaže selu kako bih mogao nastaviti dalje i usput nalazi samoga sebe. Pomagajući drugima nas lik počinje ljepše hodati i brže se kretati, ali još uvijek odustaje lagano. Glavni dio ove igre je nalaženje samoga sebe i pomaganje drugima. Ime Aria of the Soul dolazi od kombinacije muzike i duše. Kako netko ima veću dušu tako i njegova muzika je ječa i veća. I ostali kojima se pomogne igraju ulogu u muzici duše.

Program za razvoj

Sada kad imamo ideju koju razvijamo moramo odabrat u kojem programu će se izradit. Za razvoj video igre ima mnoštvo programa koji se zovu game engine*, od kojih su najpopularniji UnrealEngine*(UE) I Unity*.

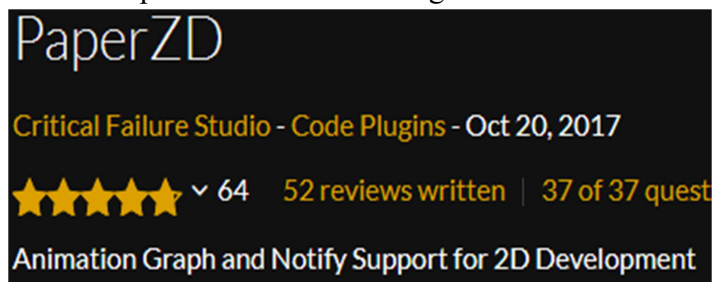
Za ovu ideju odabran je UnrealEngine, razlog za odabir UnrealEngine-a je njegov novi program UE5(Program za razvoj igra unutar Unreal engina. Najnoviji unreal engine sa neočekivanim unapređenjem izgleda video igara. Budućnost za ljepotu video igara koje se proizvode) i ljepota koja se može postići, a s osobnim znanjem o Unity-u želio sam iskusiti drugu stranu svijeta razvoja video igara.



Slika 3.: Ikone programa

S tim odabirom potrebno je znati kako postići željenu ideju unutar tog programa. Pošto UE nije game engine napravljen za 2D igre moguće je ili koristiti što je ponuđeno za razvoj za napraviti izgled 2D igre ili se može koristiti dodatak zvan PaperZD. To je dodatak za Unreal Engine koji pomaže developerima da stvore 2D igru unutar Unreal Enginea. S mnogo dodataka ovaj dodatak je moguće koristiti na mnoštvo načina, čak i napraviti igru sa 2D asetima u 3D svijetu. Više o programu biti će objašnjeno tokom ostatka razvoja.

Ova slika prikazuje dodatak PaperZD unutar Unreal trgovine.



Slika 4.: Prikaz dodatka za Unreal engine

Dokument dizajna igre (GDD)

GDD Za početak napravljen je Game Dizajn Dokument kako bi znali koji dokumenti su potrebni i za što će se koristiti. Ali to nije sve što je GDD, on posjeduje sve dijelove dizajna koje mislimo izraditi.

Priča: Ovo je priča o običnoj osobi koja je depresivna i ne želi ništa više nego sjediti u miru. Igrač je taj koji tjera tu osobu da priča sa durgima te kroz dobre interakcije glavni lik postaje manje depresivan.

Likovi: Likovi koji se nalaze u igri su: glavni lik, čovjek bez sjećanja, depresivan i tužan bez razloga, klesar koji je nema alata, depresivan kada nema alat jer nemože pomoći selu napraviti most i kuće i kovač koji je izgubio alat, depresivan bez alata koji najviše voli.

Level design: Priča se nalazi u šumi u kojoj je smješteno selo. Početak je nepoznato mjesto koje je zatamnjeno, a kasnije otkriveno da je kuća. Šuma koja je složena od više slojeva redova drveća, daje izgled hodanja kroz veliku šumu. Trava i lišće kako bi level bio unutar ekrana. Kuće sela koje su većinski iste, izrađene od istih osoba u selu. Dvije posebne kuće za likove. Most koji predstavlja cilj za igrača i kraj levela.

Gameplay: Igrivost igre nije velika, važne stvari su kretanje i interakcija. Ovo nije igra sa borbom netreba druge dodatke. Za kretanje potrebno je lagano pomicanje i brže pomicanje. U slučaju stajanja glavni lik sijedne nakon 10 sekundi. U slučaju kretanja iz pozicije sjedenja glavni lik ustaje, ta animacija traje i treba se čekati. Promjena između animacija kada igra progresira.

Slike: Potrebne su slike za sve dijelove koji su opisani unutar levela. Za drveće može jedna slika isto i za travu i lišće. Kuće su modularno napravljene kako bi imale isti izgled a unutra se može promjeniti u kovačnicu i klesarnicu. Treba i slika rijeke s mostom ali i sa srušenim mostom isto. Slika kuće glavnog lika i slika znaka kraja igre.

Muzika: Za muziku je potrebna obična lagana pjesma na koju se mogu dodati udarci koji dolaze iz sela. Treba i te udarce napraviti kao muziku.(kovač udara čekićem po metalu, klesar svojim alatom po kamenu)

Interface: Potrebni su meniji kako bi se igra mogla započeti i završiti. Isto treba i meni za pauzu kako bi se moglo stati igrati i prestati igrati.

Popis potrebnih dokumenta za razvoj

Sa odabranim game enginom i idejom treba napraviti popis dokumenta koji su potrebni za razvoj zvani aseti. Aseti su sve datoteke koje se koriste za razvoj, neki od aseta su: PNG(Portable Network Graphics format) slike, animacije*, muzika, soundeffect-ovi(Zvukovi koji se ubacuju u video igru. To mogu biti jednostavni zvukovi kao udarac, korak, Wilhelm Scream, ili mogu biti komplicirani poput muzike, ambience i glasova) i ostale datoteke.

Potrebno je imati animacije glavnog lika, animacije sporednih likova, slike pozadine, slike igrive površine i neke druge datoteke, kao muzika.

Datoteke za razvoj

Za nabavu traženih datoteka postoji par opcija. Datoteke se mogu kupiti preko interneta ili naći besplatno, ali to znači da ti aseti neće biti isti kao što su zamišljeni u ideji.

Mogu se pronaći osobe koje izrađuju asete za igre ili su artističke osobe, zaposlit ih ili ih platiti po izrađenoj datoteci. Dobra strana ovog odabira je ta što se dobije asete koji su traženi, ali najčešće je ova opcija najskuplja. Igre koje rade s budžetom (Određen broj novaca za projekt) moraju paziti da ne potroše sve novce na nabavu.

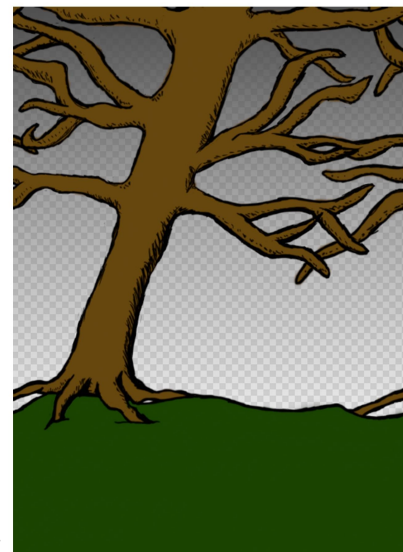
Treća opcija koja je besplatna, ali uzima najviše vremena, je da se svi aseti izrade privatno. To znači da se oslanjamo na vlastite osobne vještine te ako je igra zahtjeva nešto novo, ili loše izgleda, može i mora biti osobno popravljeno.

Za ovaj rad odabrana je treća opcija. Svaki aset koji je unutar ovog programa je osobno izrađen.

Datoteke slika

Slike su izrađene u Photoshop* programu kako bila prazna pozadina i slike mogu biti izrađene modularno. Koriste se posebni izbori za izradu i stvara se u layerima. Koristi se mnoštvo alata, ali najčešće korišteni alati su: BrushTool, EraserTool, EyedropperTool, ColorPickerTab, LayerTab, SymmetryOption i ZoomTool.

Za pozadinu ove igre odlučilo se na izgled tamne šume, s tom idejom izrađena je slika koja se može ponoviti više puta i koja će stvoriti iluziju šume i pozadine.



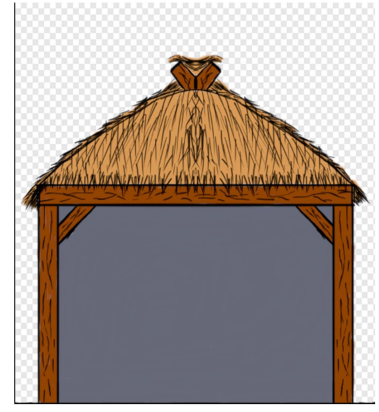
Slika 5.: Isječak pozadine.

Za grad koji je potreban za priču odlučeno je slikanje na modularni način izrade. Prvo je napravljen vanjski izgled kuće koji će biti isti za svaku kuću u gradu.

Tada je dodana pozadina kao drugi dio te slike.



Slika 6.: Dio kuće koji se ponavlja.



Slika 7.: Pozadina kuća.



Slika 8.: Prednji dio kuća.

I za kraj sam dodana je prednja strana zatvorene kuće kako bi mogli dodati kuće bez značaja za priču.

Ove slike prikazuju sve što je izrađeno u Photoshopu za elemente igre.



Slika 9.: Izgled kovačnice.

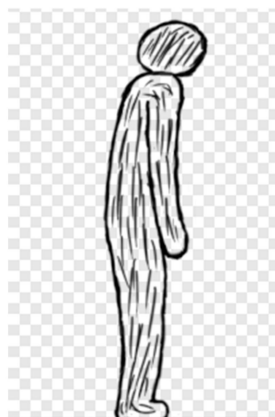
Kuće koje imaju značaj za priču su otvorene, te su za svaku napravljena tri dodatka. Dodatak za pozadinu kako kuća nebi izgledala prazno. Glavni dio kuće koji označava što ta kuća predstavlja. Te osoba koja u toj kući živi, sa animacijom pokreta.



Slika 10.: Izgled klesara.

I zadnji komplet slika je glavni lik kojem trebaju animacije za sijesti, dignut se, hodat i hodat umorno. Za to je potrebno napraviti početni izgled lika.

Smjestit ga u sve pozicije koje su potrebne.



Slika 11.: Izgled lika.



Slika 12.: Pozicija sjedenja.



Slika 14.: Animacija hodanja.

I iz tih pozicija se animira što treba prikazati.

Ovdje su prikazane slike koje se koriste u animiranju glavnog lika.



Slika 13.: Animacija ustajanja.



Slika 16.: Oznaka kraja demo igre.

Za kraj dodaje se izmjene kao što je rečeno da se ideje mjenjaju. Dodaje se sliku rijeke s mostom na kraju grada te sliku stupa koji označava da je igra u demo* stanju. Slike su dodatka koji su korišteni u igri.



Slika 15.: Izgled rijeke i mosta.

S time je gotov razvoj svih slika koje su potrebne za izradu igre.

Datoteke muzike

Muzički odabir se svodi na kupovinu muzike, plaćanje muzičara, nalaženje RoyaltyFree(Stvari koje su RoyaltyFree su slobodni za korištenje jer ne pripadaju jednoj osobi, već svima koji ih žele koristiti. Drugim riječima asseti koji su besplatni za korištenje) muzike ili izradu svoje muzike. Ponovo je odabrano stvaranje svoje muzike.

Za muziku je odabran jednostavni program za izradu lagane muzike zvan BoscaCoil(Program u kojem se stvara muzika preko MIDI tipkovnice. Jednostavan program za slaganje muzike).

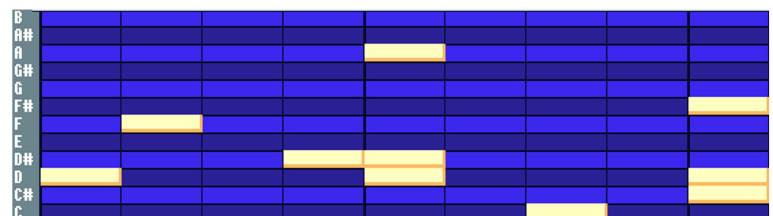
Sa limitiranim znanjem muzike kreće se u sklapanje neke vrste pozadinske muzike koja ima smisla sa ostatkom igre.

Izrada muzike bez znanja o tome kako složiti muziku spada na pokušavanje i probavanje. Nakon dugog postupka izrade složena su tri identična zvuka sa dva dodatka udarca koji se pojavljuju u priči.

To bi trebalo stvoriti iluziju napredovanja i promjene u svijetu. Ove slike prikazuju izgled programa BoscaCeoil.



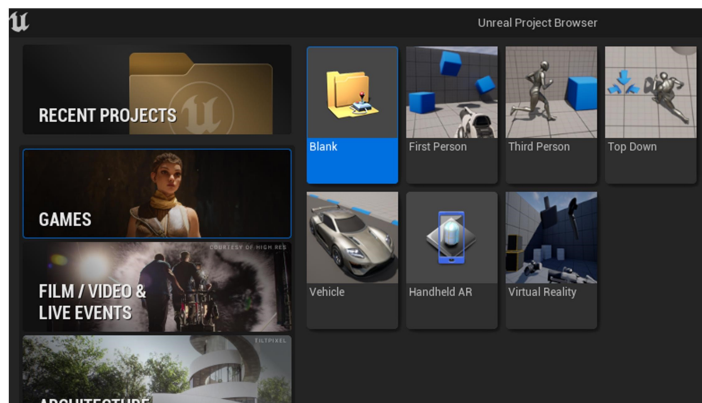
Slika 17.: Program za muziku.



Slika 18.: Izgled slaganja muzike.

Programiranje

Kako bi se započelo programiranje otvara se program koji je odabran. Unutar UE5 postoji par opcija za započet projekt, može se odabrati igru, filmove i mnoge druge opcije od koji će se odabrati igru i to prazan projekt kako bi se počelo iz nule. Na taj način je stvorena prazna ploha na kojoj se može izgraditi novi projekt umjesto da se doraduje na tuđoj strukturi.



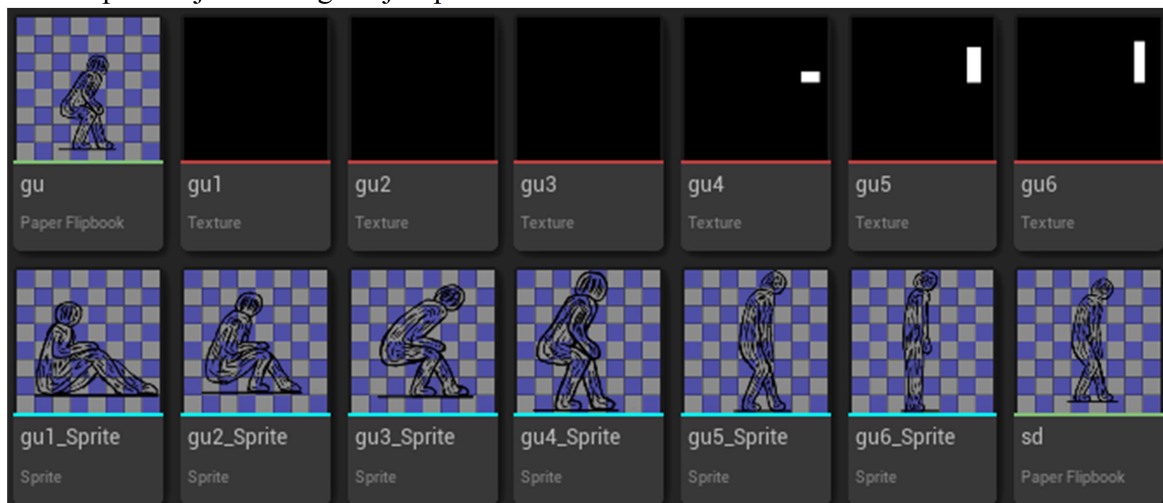
Slika 19.: Odabir projekta u UE.

Ovo je slika UnrealEngina i otvaranja novog projekta.

S time se otvara program koji je prazan, zato mu se dodava asete koji su pripremljeni.

Nakon što se dodaju svi aseti mora se pretvoriti sve PNG slike u sprite asete koje se onda pretvaraju u flipbook(Asset u UE5 unutar kojeg se izmjenjuju nekolicina spriteova kako bi stvorili animaciju). U slučaju da se radi sa PixelArt metodom(Tip crtanja slika, koristi kockastu strukturu(grid) kako bi napravio izgled pikseliziranog crteža. Stil crtanja koji se koristi u video igrama čak i danas kada odlična grafika prevladava) ovdje bih trebalo pretvoriti PNG prvo u Paper2D(dodatak unutar UE5 s kojim stvaramo dijelove asseta kako bih mogli napraviti 2D igru, ili makar 2D asete) slike kako se nebi izgubilo željene izgleda. U ovom slučaju to nije potrebno pa se može odmah raditi sa tim slikama. Isto tako ako je PixelArt za pozadinu i level tada se izrađuje tileset(Asset unutar UE5 s kojim stvaramo levele. Može služiti za slikanje izgleda levela, pozadine ili ostalih dijelova igre) kako bi se moglo slikati izgled levela pomoću isječka naših slika. Pošto u ovom program se ne radi sa PixelArt-om preskače se taj korak i nakon što se sve pretvori u spriteove, slaže se level pomoću njih.

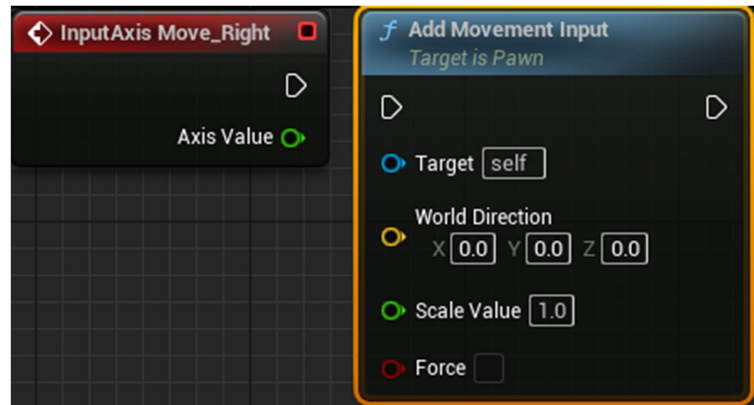
Ova slika prikazuje kako izgledaju Sprite-ovi unutar UE5.



Slika 20.: Stvaranje spriteova.

Nakon što se složi level, ili makar jedan dio na kome će se testirati, kreće se u izradnju lika. Napravi se blueprint (Aset unutar UE5 s kojim možemo upravljati i kojeg možemo programirati) za glavnog lika i ubaci mu se jednu od animacija koje su napravljene pomoću flipbooka. Na isti taj blueprint se zalijepi i kamera kako bi kamera pratila našeg lika. Tu se naprave male promjene sa svjetlom, te nastavi se na slijedeći dio, programiranje kretanja.

Kako bi se programiralo kretanje lika ponovo se otvara blueprint lika te se odabire njegov EventGraf (Dio postavka unutar blueprinta. U event grafu radimo većinu programiranja blueprinta), unutar kojeg se dodaje sve što je potrebno da bi se glavni lik kretao kako je zamišljeno. Ova slika prikazuje blueprintove potrebne za kretanje.

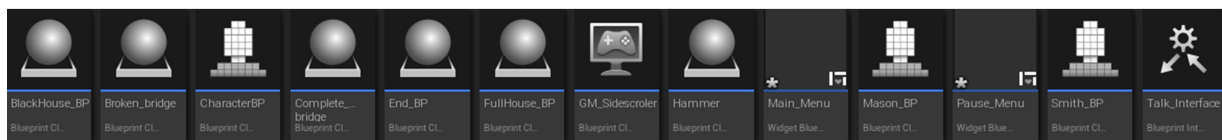


Slika 21.: Programiranje kretanja.

Dodana je i jedna ploha u pozadini kako pozadina nebi bila prazna, na tu plohu zalijepljen je material određene boje kako bi stvorio svjetliju pozadinu za igru.

Sada kad je završen level i djelomično napravljen lik koji može hodati, mora mu se dodati osobe s kojima će i pričati. Zbog toga se radi još dva blueprinta za klesara i kovača. Tu se dodaje i njihove interakcije sa glavnim likom te promjene animacije. Kad su gotovi NPC (Non-playable-character) lik unutar igre kojeg igrač ne kontrolira. Najčešće kontroliran sa AI) likovi dodaje se još par interakcijskih dijelova koji su potrebni za priču.

Slika prikazuje izgled blueprintova unutar UE5

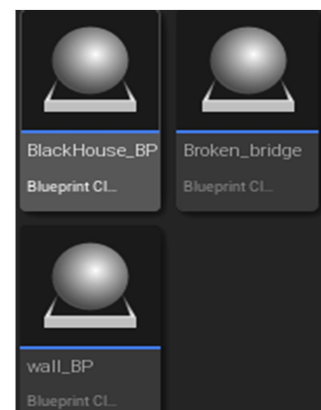


Slika 22.:Blueprintovi.

Još će se dodati par slika blueprintova koje će biti objašnjene kako rade. Naravno ne smije se prikazati kako su ti blueprintovi spojeni, ali se smije pokazati koji su blueprintovi korišteni.

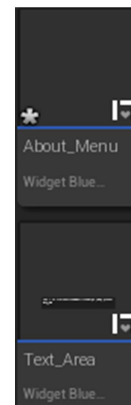
Korišteni Blueprintovi su:

-actor je blueprint klasa predmeta koji ima funkciju unutar levela. Odnosi se na bilo koji objekt ili entitet unutar igračkog svijeta. Ova široka kategorija uključuje likove, predmete, pozadinu i još mnogo toga. Ovaj blueprint omogućuje developerima definiranje ponašanja, svojstava i interakcija. Na primjer možete stvoriti Blueprint glumca za upravljanje kretanjem, animacijama i interakcijama glavnog lika u igri.



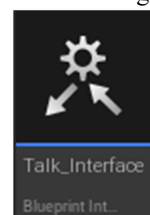
Slika 23.:Actor.

-widget je blueprint klasa koja se koristi za izradu korisničkih sučelja (UI) unutar igre. Widgeti su bitni za dizajniranje i implementaciju izbornika, prikaza na ekranu (HUD), elemenata UI-a unutar igre i drugih grafičko korisničkih sučelja. Pružaju vizualni prikaz informacija, kontrole ili interaktivnih elemenata koje igrači mogu vidjeti. Widgeti su visoko prilagodljivi i mogu uključivati gumbe, tekstualna polja, slike i još mnogo toga.



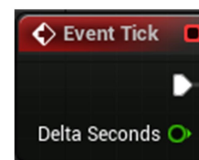
Slika 24.:Widget.

-interface je blueprint klasa koja pruža način za definiranje skupa funkcija ili metoda koje mogu implementirati različiti glumci ili klase. Ova funkcionalnost izuzetno je korisna za uspostavljanje zajedničkih ponašanja ili sposobnosti koje različiti glumci mogu dijeliti, bez potrebe da nasljeđuju od zajedničke nadređene klase. Na primjer, možete stvoriti sučelje za interaktivne objekte poput vrata, omogućujući različitim glumcima da implementiraju to sučelje i dosljedno reagiraju na igračke interakcije. Sučelja poboljšavaju modularnost i ponovnu upotrebu koda. Ukratko, glumci i sučelja su neophodni alati unutar UE, koji omogućuju developerima da jednostavno kreiraju dinamične i interaktivne igračke svjetove.



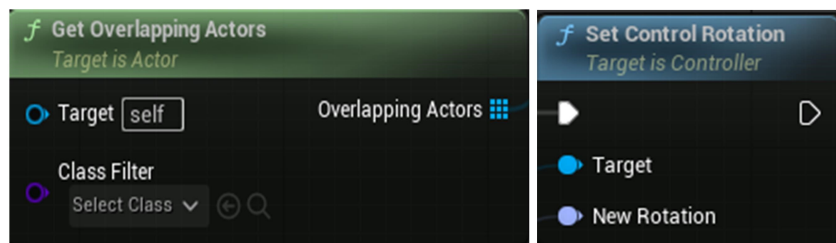
Slika 25.:Interface.

-event je blueprint koji se koristi kao ključni okidač u logici igre. Oni reagiraju na specifične događaje ili akcije unutar igre. U igri, event, može se koristiti za detekciju kada igrač pritisne gumb, što pokreće niz akcija, poput stvaranja actora i promjene izgleda.



Slika 26.:Event.

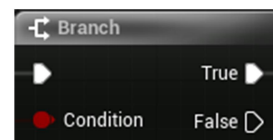
-set i get su blueprints temeljni za upravljanje varijablama unutar Blueprint skripti. Oni omogućuju dodjeljivanje novih vrijednosti varijablama ili dohvaćanje trenutnih vrijednosti varijabli. U igri sa



Slika 27.:Set i Get.

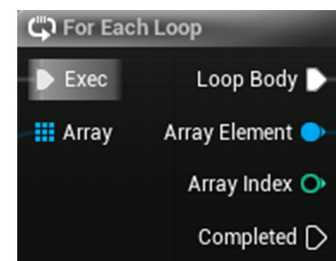
upravljanjem resursima, možete koristiti set za oduzimanje resursa kada igrač mijenja nešto i get za provjeru imaju li dovoljno resursa za tu promjenu.

-branch je blueprint koji služi kao točka donošenja odluka unutar Blueprint logike. Evaluira se određeni uvjet i usmjerava tijek skripte ovisno o tome je li taj uvjet istinit ili lažan. U avanturističkoj igri, branch može provjeriti je li igrač odabrao određeni odgovor u dijalogu, usmjeravajući priču prema tome odabiru.



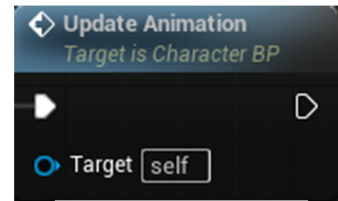
Slika 28.:Branch.

-for je blueprint koji je neophodan za obradu ponavljajućih zadataka ili iteriranje. U strateškoj igri, možete koristiti for za prolazak kroz jedinice koje igrač kontrolira i provjeru jesu li unutar dometa. Za svaku jedinicu unutar dometa, možete pokrenuti određenu akciju.



Slika 29.:For.

-update je blueprint koji neprestano prati promjene tijekom vremena ili unos igrača. To je ključna komponenta za stvarno-vremenske sustave unutar Blueprinta. U igri utrka, update može snimati unos kontrolera, omogućujući igračima upravljanje vozilom i kontrolu nad ubrzanjem ili usporavanjem.



Slika 30.:Update.

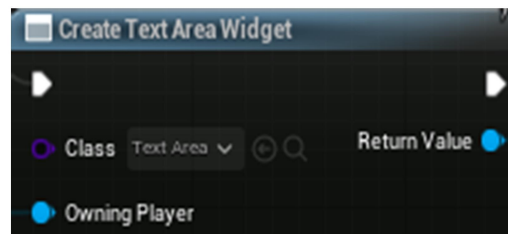
-bool set je blueprint koji mijenja stanje varijabli unutar Blueprinta. Na primjer, promjena ako je glavni lik na podu ili u zraku.



Slika 31.:SetBool i Compare.

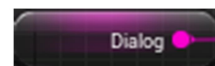
-compare je blueprint koji procjenjuje odnos između dviju varijabli, određujući jesu li jednake, različite, veće ili manje od druge. U slučaju kretnje lika provjerava ako je brzina veća od nule

-create je blueprint koji je ključan za stvaranje novih objekata ili resursa tijekom igre. U igri, oni mogu instancirati platforme kada igrač dosegne određene lokacije, dodavanje neprijatelja, stvaranje Pickup-a ili stvaranje novog UI elementa.



Slika 32.:Create.

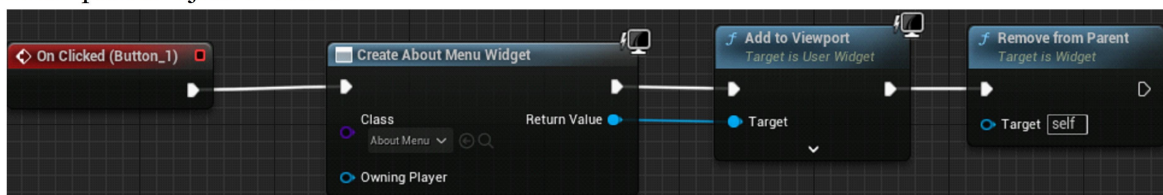
-variable je blueprint varijable koji dohvaća trenutnu vrijednost varijabli definiranih unutar Blueprinta. Na primjer, u RPG-u, ovi čvorovi mogu pristupiti zdravstvenim bodovima igrača, koji se mogu prikazivati u UI elementu trake sa zdravljem i koristiti za mehanike kao što je izračunavanje štete od neprijatelja.



Slika 33.:Variable.

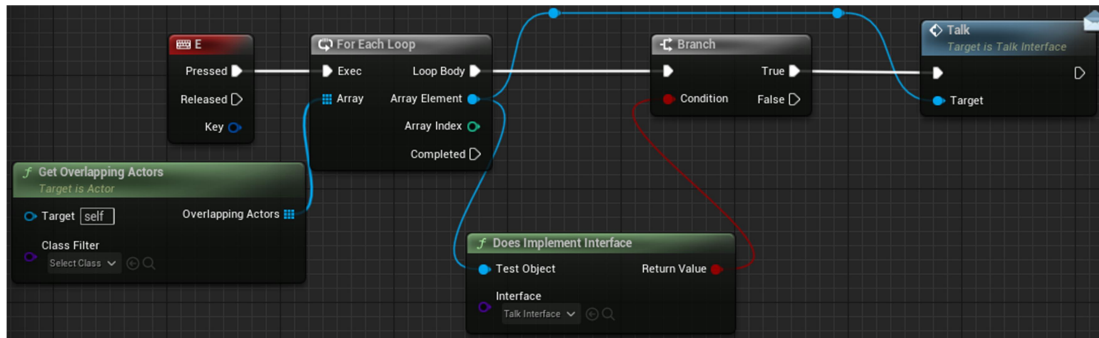
I mnogi drugi blueprintovi još postoje unutar Unreal engina. Svaki blueprint je kolekcija koda koji je dobro napravljen kako bi se mogao koristiti sa bilo kakvim varijablama. Oni pretvaraju pisanje koda u jednostavniju igru puzla sa blokovima koda.

Ovo su neke od kombinacija blueprintova koje se nalaze u ovom radu. Probat ću ih objasniti ukratko prateći njihove slike.



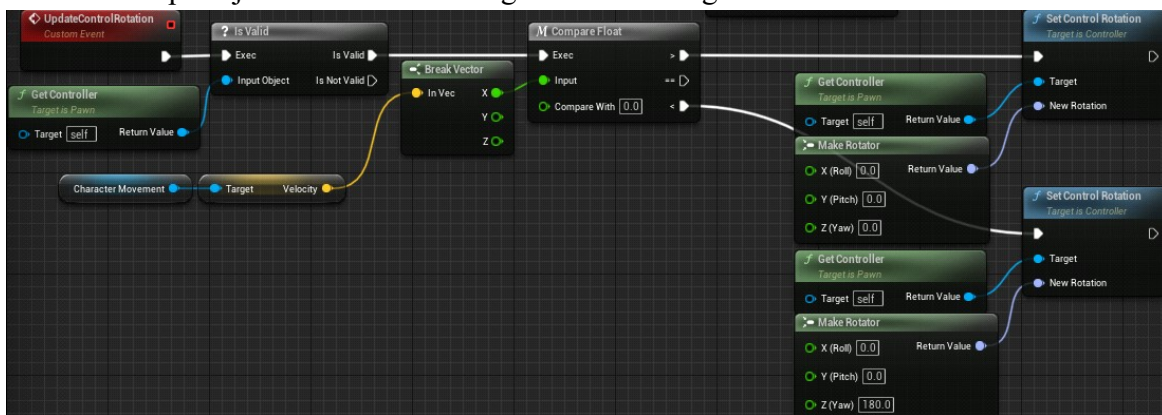
Slika 34.:Blueprint kombinacija za stvaranje menija.

Na ovoj slici je prikazan proces stvaranja i uništavanja UI elemenata. U ovome slučaju to je promjena iz menija u drugi meni. Unutar prvog menija dodan je event blueprint koji se poziva na kliknuti gumb koji se nalazi unutar menija. Taj event stvara sa create blueprintom novi UI meni te ga sa add BP dodaje na ekran. I za kraj uništava stari UI meni sa remove BP.



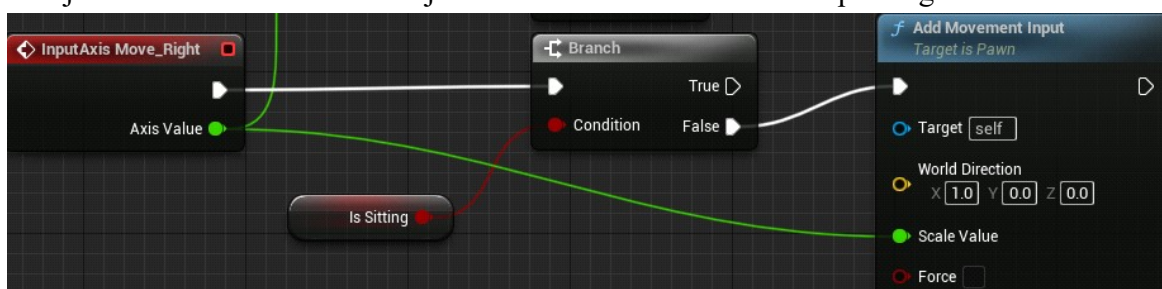
Slika 35.:Blueprint kombinacija za pričanje s NPC.

Na ovoj slici je prikazan proces interakcije koji se pokrene eventom kada je pritisnuta tipka E i glavni lik je u blizini NPC-a. Koristimo for kako bi mogli promijenit što će NPC govoriti, branch kako bi provjerili da li već govorimo te update na interface kako bi pokrenuli govor. Ostali get BP su tamo da provjerimo ako smo u krugu NPC-a i ako govorimo.



Slika 36.:Blueprint kombinacija za promjenu orijentacije.

Na ovoj slici je prikazan proces promjene smijera kretanja u slučaju okreta u suprotnu stranu. Važni dio ovog procesa je provjera compare-om u koju se stranu kreće glavni lik, i postavljanje set-om na rotaciju koju smo odredili u produženom get blueprintu unutar kojeg smo dohvatili našeg lika i silovito mu promijenili orijentaciju. isto tako smo morali i slomiti vector kretnje lika na varijable kako bi dobili točnu vrijednost nevažno dali se on kreće po drugim osima.



Slika 37.:Blueprint kombinacija za kretanje lika.

Na ovoj slici je prikazana kretanja glavnog lika. Važni blueprintovi kod kretnje lika su dobava unosa kretnje, provjera kroz branch ako glavni lik sjedi, što bi značilo da se nemože kretati, i dodavanje kretnje tom liku kroz add movement input, bez čega se lik nebi mogao niti pomaknuti.

UE Blueprints je moćan vizualni sustav skriptiranja unutar UE, primarno dizajniran za razvoj igara. Ovi blueprints omogućuju developerima, bili oni početnici u programiranju ili iskusni programeri, da stvaraju kompleksnu logiku igre i interaktivne elemente bez da napišu jedan redak koda. Povezivanjem predefiniраниh blueprintova i logičkih blokova unutar grafičkog sučelja temeljenog na blueprint-ima, dizajneri igara i developeri mogu brzo izraditi prototip i implementirati mehaniku igre, AI ponašanja, korisnička sučelja i još mnogo toga. Na primjer, dizajner igara može koristiti blueprints za stvaranje sustava otvaranja i zatvaranja vrata u igri kada ih igrač klikne. Osim toga, blueprints se mogu koristiti za dizajniranje kompleksnih AI ponašanja, poput neprijatelja koji patroliraju određenim područjem ili reagiraju dinamično na akcije igrača. UE Blueprints omogućuju širokom spektru kreativnih umova da svoje ideje pretvore u stvarnost u svijetu razvoja igara, čineći ga vrijednim alatom, kako za početnike tako i za iskusne profesionalce.

Ali naravno to nemože biti sve, prave igre koje su napravljene pomoću blueprintova isto tako koriste i svoj kod. Blueprints su odlični za početak, ali nažalost im fali finisa. Oni mali detalji koje možemo popraviti samo u kodu i moramo posebno napisati. Kako bi igra imala točno što želi, mora imati i pisani kod kako bi popravila nedostatke blueprintova.

I s time je završen demo ove igre. Naravno da za punu igru ovdje nije kraj, i to će biti objašnjeno kroz ostatak ovog projekta.

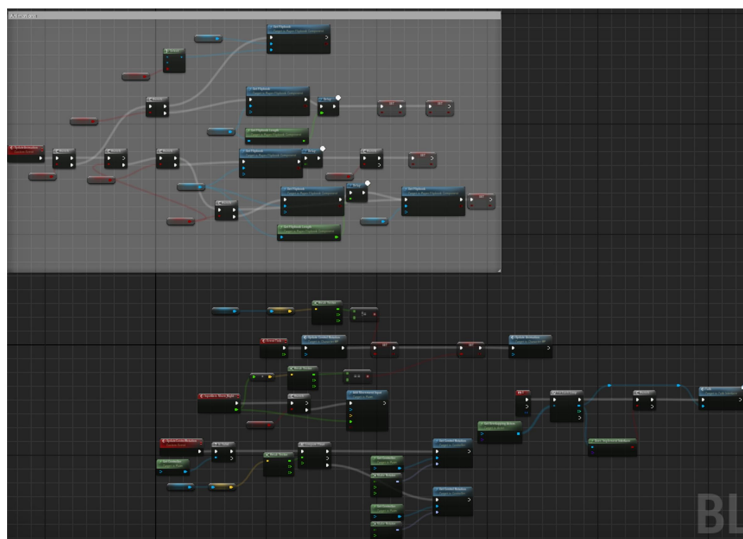
Testiranje i promjene

Nakon što je napravljen demo koji je funkcionalan, kreće se sa testiranjem glavnog lika i promjenama koje su napravljene tokom ovog projekta. Za početak programirane su sve animacije glavnog lika te dobro su testirane kako bi sve animacije radile.

Kada je završen glavni lik, dodane su i interakcije koje NPC likovi imaju na svijet u kojem se nalaze.

Dodani su i ukrasi na level kako bi igra izgledala dobro. Te su napravljene neke promjene scene. Za kraj je dodan i menu kako bi igra više izgledala na igru.

Ova slika prikazuje kako izgleda razvijeni Event graf.

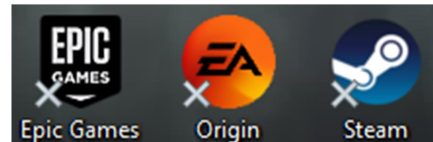


Slika 38.:Event graf.

I naravno na kraju ove faze kako bi se testiralo ako svaki dio ove igre radi, odigrat će se cijeli demo više puta na više načina. Ova faza je najčešće nešto što rade Game tester^{*}, osobe koje igraju igru prije nego izađe kako bi isprobali elemente igre i javili probleme developerima. Oni testiraju što je više moguće, diskretno, kako bi igra bila spremna za prodaju, nakon što developer popravi sve što testerⁱ nađu.

Prodaja i distribucija

I kada je sve gotovo i igra je izrađena, tada dolazi marketing i prodaja. To se dešava na platformama poput Steam, Epic Games, Origin, GOG. Te platforme prodaju i distribuiraju igru ljudima oko svijeta.



Slika 39.:Distributeri.

Ovdje su na slici prikazane ikone od nekih platforma koje distributiraju igre.

Rasprava (Motivacija i poruka programerima)

Nakon što sam završio ovaj rad okrenuo sam pogled na rad koji sam napravio, te s time došao do zaključka koji je zadatak ovog rada.

Unutar ovog projekta proveo sam više vremena tražeći što želim dodat i učeći programe nego što sam stvarao ono što sam htio. Zbog mogjeg odabira da napravim asete sam i ne kupim ništa, imao sam više problema nego ikada. Stvarati slike u photoshopu bez nekakvog dodatka i specijalnog znanja je jako teško.

Slike najčešće nisu ispale kakve želim i morao sam ih puno napraviti iako sam probao smanjiti stvarajući modularno. Stvaranje muzike je stvarni problem za nekoga tko ne zna koja nota kakav zvuk stvara i nema smisao za ritam. Koristeći jednostavan program olakšava, ali ne dovoljno. I da niti ne počinjem sa svim dijelovima UnrealEngine-a. Od znanja kako dodati dijelove do toga kako upravljati njima, svaki dio ovoga rada bio je težak.

Ali iz drugog pogleda, ovaj projekt ne zahtijeva ništa drugo nego volju i kompjuter, malo imaginacije i ako je potrebno talent. Stvaranje slika je samo imaginacija i igranje, kao što smo se igrali u program Paint dok smo bili mali. Muzika je isto lagana. Tu nema puno za raditi, samo stvoriti nešto što nam se sviđa lupajući po notama. Kao djeca smo se igrali sa ksilofonom i stvarali muziku, BoscaCoil nije ništa drukčiji, samo je digitalan. Cijeli razvoj video igre možemo naći na internetu. Sve što želimo napraviti može se naći, netko iz je već napravio. Sve što je potrebno je znanje što želimo dodati i kako napraviti da izgleda kako mi želimo.

I s tim zaključkom možemo reći da je svrha ovog rada postignuta, razvoj video igra je težak posao koji zahtijeva puno, ima puno problema i težak je na osobe koje razvijaju. Ali isto tako možemo reći da je cilj rada isto postignut, razvoj video igre nikad nije bio lakši. Razvoj zahtijeva skoro ništa od osoba koje razvijaju, ne zahtijeva nikakav buđet, sve je dostupno, ne traži znanje i sada ima i putokaz u obliku projektnog rada.

S toga želim, svima koji žele napraviti svoju video igru, da ih ne zaustavlja veličina projekta, nego da uzmu svoju ideju i isprobaju razviti video igru. Nije važno kako će ta igra izgledati, ono što je važno je da će se još jedna idea oživiti u obliku video igre. I nikad se ne zna tko će razviti igru koja će postati popularna. Kao i FlappyBird, i Pac-Man - priča o gladnom čovjeku koji zna kako je to biti bez hrane i samo želi da svi mogu imati puno, poput njegovog lika Pac-man-a.

Zaključak

Ovaj projekt je počeo sa idejom koja je evoluirala tokom ovog rada. Odlučeno je što se želi napraviti sa tom idejom i odabrao se tip igre 2D Sidescroller. Kad je ideja bila spremna odabralo se i Game Engine, i to UnrealEngine 5. Za ovaj rad nije bilo budžeta pa se odlučilo sve asete napraviti privatno. Za sve artistske dijelove iskorišten je program Photoshop i puno opcija koje su bile ponuđene. Za muziku odlučeno je izrada na programu BoscaCoil te složeni su neki aseti. Svi aseti su tada ubačeni u program UE5 te sa dodatcima paper2D i paperZD, te Blueprint načinom programiranja krenuto je u izradu igre. Sa puno rada i promjena napravljen je projekt koji se može zvati igrom. Taj projekt je dobro testiran i popravljen kako bi stvarno izgledao na igru. Ta igra je onda zapakirana i s time je razvijena video igra koja se htjela razviti.

Napravljena je igra po ideji koja je bila zadana, u program koji je bio odabran i na način na koji je trebala biti razvijena. Iskorišteno je nekolicina programa za puno drukčijih zadataka poput slika i muzike. Svi aseti su izrađeni i stavljeni u game engine, te složeni u izgled video igre. Programirano je sa Unreal engine 5 blueprintovima i izrađena je igra koja se može pokrenuti i igrati. Korišteno je većina faza razvoja video igre i napravljeni su skoro svi dokumenti za razvoj. Dodan je i ispunjen osobni zadatak rada. S ovim je završen zadatak završnog rada i postignut razvoj računalne (video) igre.

Aria of the soul

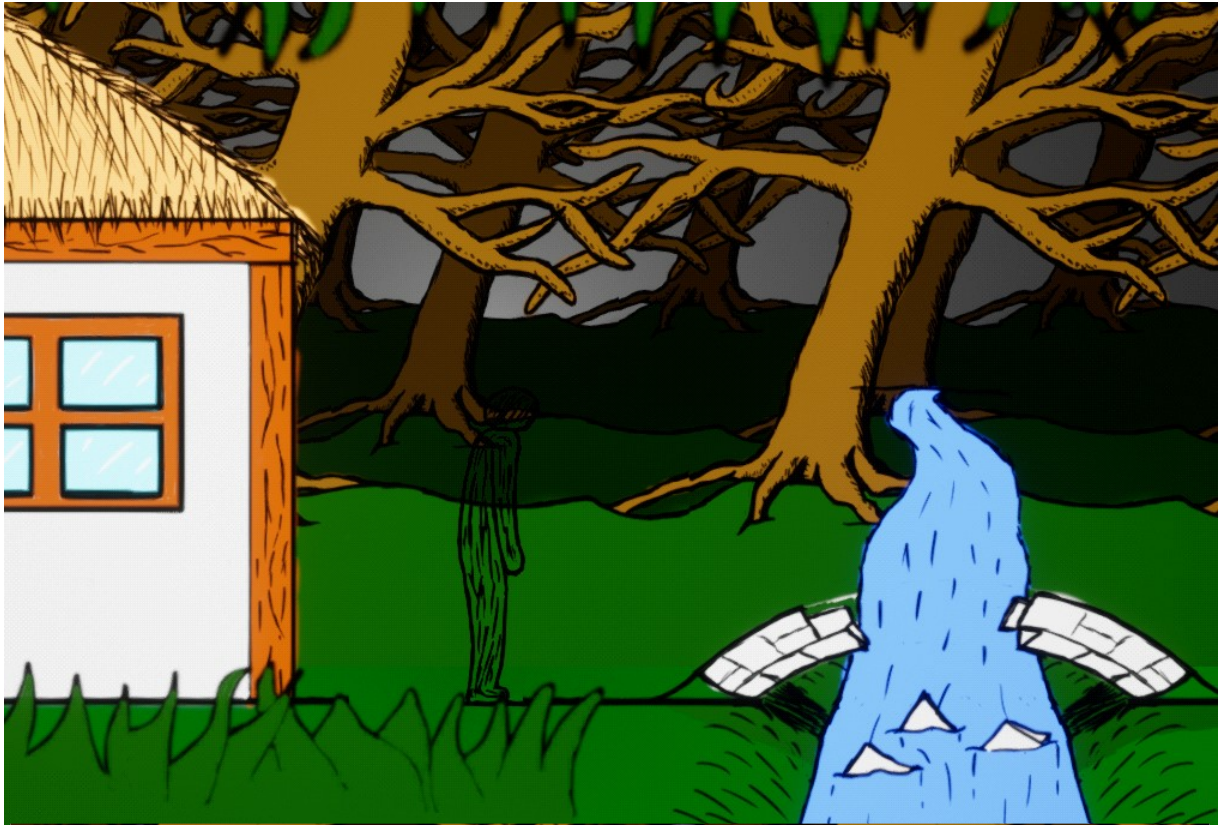
Restart

Continue

About

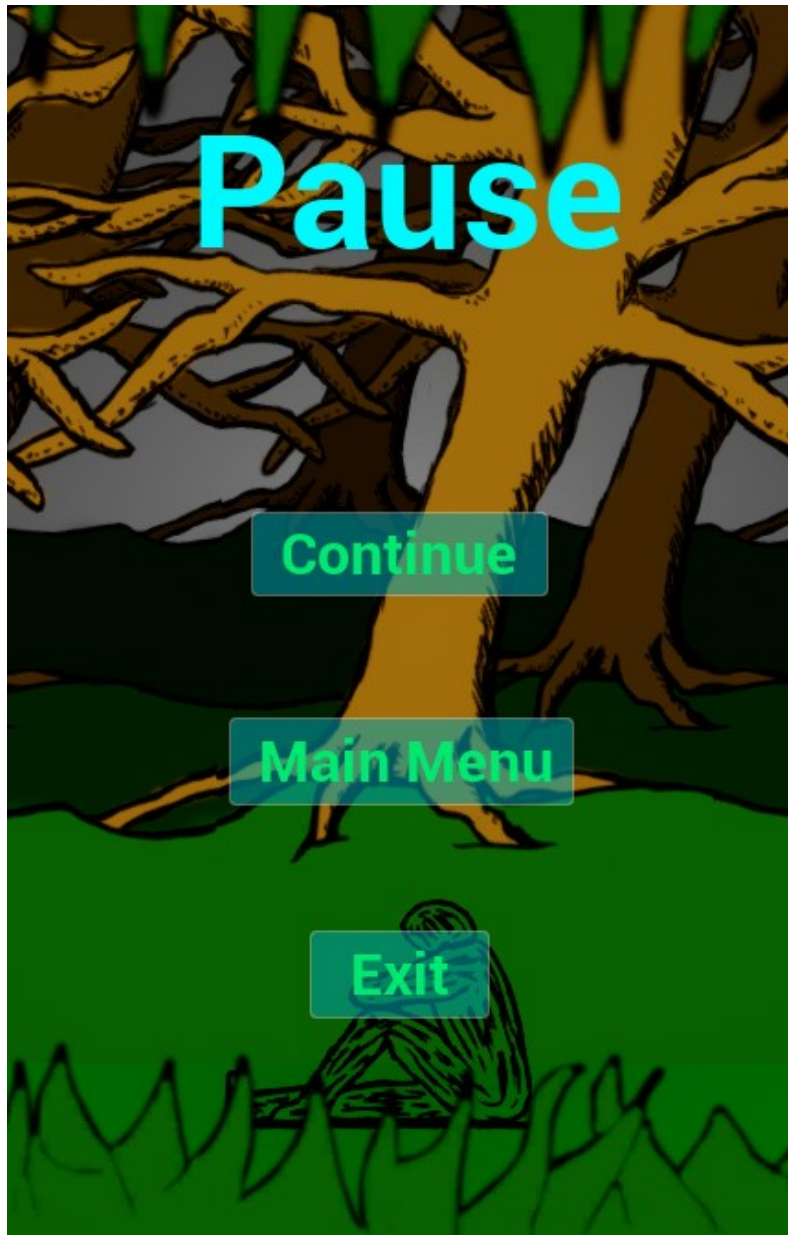
EXIT











Slike 40. – 48.: Prikaz izgleda video igre.

Opis oznaka

Animacije* - Animacije mogu biti više drukčijih vrsta. U 3D igrama animacije su najčešće fluidne i bez gubitka pokreta dok u 2D igrama animacije dolaze u par vrsta:

-Frame By Frame animacija – animacija koja za svaki frame ima drukčiju sliku koju koristi;

-Film animacija – animacija koja koristi 12 do 24 slika u sekundi i s time proizvodi fluidnu animaciju likova i objekata;

-Stop Motion animacija – animacija koja koristi broj slika koji je developer želio koristiti kako bih prikazao svoju ideju. Ovakva animacija ima puno drukčijih načina rada, držanje jedne slike par sekundi ili zaguravanje puno slika u jednu sekundu, svaki način korištenja stop motion animacije ima svoju svrhu te bolje prikazuje originalnu ideju.

Aria of the Soul* - Video igra koju razvijamo tokom ovog rada. Igra je tipa Sidescroller i priča priču o glavnom liku i svijetu u kome se nalazi. Napravljena je sa pogledom na utjecaj depresije i njen utjecaj na svijet oko glavnog lika.

Binding of Isaac* - Video igra koja prikazuje priču o djetetu koje želi pobijeći od svoje majke, koja je čula glas boga i on je naredio da ubije svoje dijete kako bi iskazala svoju vjeru.

Demo* - Verzija programa u kojoj nije dozvoljen pristup cijelom programu. Ova verzija koristi se u mnogo svrha, prikazivanje programa na eventovima, prezentiranje programa i ideja, zaštita koda i ideje, ili strateški kako bi nakon što korisnici isprobaju program i primjete da im se sviđa ili žele više opcija, bilo im dozvoljeno da kupe punu verziju programa koja je kompletirana sa svim opcijama koje su i trebale biti u tome programu.

Fortnight* - Video igra tipa Battle royale koja uzima 100 osoba i postavi ih u borbu gdje može biti samo jedan pobjednik. Napravljena je s izgledom crtića, te isto tako drži do crtića u borbi.

Game engine* - Program u kojemu se razvijaju video igre. Svaki Game engine je drugačiji, ima svoje prednosti i mane te neki game engine su razvijeni od strane kompanija za igre kako bi bili kompatibilni za njihove igre i ideje, te da lakše proizvedu te igre. Neke kompanije su i propale pokušavajući izraditi game engine za svoje igre.

Game tester* - Osobe u svijetu proizvodnje video igara koje su odgovorne za testiranje igre te nalaženje grešaka unutar igra. Najčešće to zovemo game breaking jer oni pokušavaju razbiti normalan način igre kako bi pronašli probleme koji se mogu stvoriti.

Minecraft* - Video igra u kojoj glavni lik ima moći teraformiranja, igra je tipa sandbox u kojoj igrač može raditi što god poželi. Napravljena je s kockastim svijetom što pretvara igru u laganu igru građenja.

Photoshop* - Program za izradu i uređivanje slika. Ovaj program ima jako puno opcija, iako je predstavljen kao program za uređivanje, u ovom projektu je iskorišten kao program za stvaranje na praznoj površini kako bih lagano pretvorili te slike u dodatke za igru.

Pubg* - Video igra tipa Battle royale koja uzima 100 osoba i postavi ih u borbu gdje može biti samo jedan pobjednik. Napravljena je s realističkim izgledom te isto tako drži do realizma u borbi.

Shaolin vs Wutang 2* - Nastavak prve igre Shaolin vs Wutang! Ova igra je indie igra napravljena kao ljubavno pismo borilačkim vještinama i klasičnim kung fu filmovima.

Unity* - Program za razvoj video igara. Ovaj game engine je univerzalan na skoro svim tipovima igra i dobar je početak za indie developera. Ima puno ljudi koji izrađuju igre u njemu, te s time ima i puno informacija koje mogu pomoći početnicima kod izrade igre.

Unreal Engine* - Program za razvoj video igara koji se nalazi na platformi Epic Games. Ovo je game engine koji smo odabrali za projekt. Ovaj game engine poznat je po svojim nevjerojatnim izgledama igara napravljenih u njemu, te nezamislivim animacijama.

Witcher* - Knjige autora Andrzej Sapkowski koje opisuju priču o Geraltu iz Rivije, Witcher, koji lovi čudovišta.

Popis slika

1. Slika rezultata igre Assassin's Creed Valhalla na stranici metacritic kako bi prikazali primjer dobre igre.
2. Slika rezultata igre Lord of the rings: Gollum na stranici metacritic kako bi prikazali primjer loše igre.
3. Slika ikona dva najpopularnija programa za programiranje video igara.
4. Slika dodatka PaperZD koji možemo koristiti unutar Unreal Engine programa kako bi olakšali izradu 2D igre.
5. Slika dijela pozadine od koje slazemo pozadinu.
6. Slika vanjskog dijela kuće koji se nalazi na svakoj kući.
7. Slika koja prikazuje kako izgleda pozadina u kucama kojima vidimo unutrašnjost.
8. Slika koja prikazuje kako izgleda prednji dio kuća kojima ne vidimo unutrašnjost.
9. Slika koja prikazuje kako izgleda unutrašnjost kovacnice.
10. Slika koja prikazuje kako izgleda unutrašnjost kuće klesara.
11. Slika koja prikazuje glavnog lika ovog projekta.
12. Slika koja prikazuje glavnog lika u poziciji sjedenja.
13. Slika koja prikazuje kako bi glavni lik izgledao u animaciji ustajanja.
14. Slika koja prikazuje kako bi glavni lik izgledao u animaciji hodanja.
15. Slika koja prikazuje rijeku i most koji se nalaze u projektu.

16. Slika koja prikazuje oznaku kraja demo igre.
17. Slika koja prikazuje program za izradu muzike.
18. Slika koja prikazuje primjer izgleda muzike unutar programa Bosca Ceoil.
19. Slika koja prikazuje kako izgleda odabir projekta unutar programa Unreal Engine 5.
20. Slika koja prikazuje spritove unutar Unreal engina. Ova slika prikazuje spritove za dizanje i sjedenje glavnog lika.
21. Slika programiranja kretanja. Ova slika prikazuje kako izgledaju blueprint klase unutar event grafa.
22. Slika blueprintova unutar Unreal engina 5. Ova slika prikazuje neke od blueprintova koji su korišteni u ovom radu.
23. Slika koja prikazuje kako izgledaju blueprinti actora u UE.
24. Slika koja prikazuje kako izgledaju blueprinti widgeta u UE.
25. Slika koja prikazuje kako izgledaju blueprinti interfecea u UE.
26. Slika koja prikazuje kako izgleda event blueprint u UE.
27. Slika koja prikazuje kako izgledaju set i get blueprinti u UE.
28. Slika koja prikazuje kako izgleda branch blueprint u UE.
29. Slika koja prikazuje kako izgleda for blueprint u UE.
30. Slika koja prikazuje kako izgleda update blueprint u UE.
31. Slika koja prikazuje kako izgledaju boolset i compare blueprinti u UE.
32. Slika koja prikazuje kako izgleda Create blueprint u UE.
33. Slika koja prikazuje kako izgleda variable blueprint u UE.
34. Slika koja prikazuje kombinaciju za stvaranje menija unutar UE.
35. Slika koja prikazuje kombinaciju za interakcije unutar UE.
36. Slika koja prikazuje kombinaciju za promjenu orijentacije unutar UE.
37. Slika koja prikazuje kombinaciju za kretanje ili pomicanje unutar UE.
38. Slika event grafa. Ova slika prikazuje kako može izgledati event graf unutar blueprinta, te prikazuje da i blueprint programiranje može biti kompleksno.
39. Slika icona aplikacija distributera. Distributeri prodavaju igre kroz aplikacije, ovo su neke od njih.
40. Screenshot igre, Title screen početni izgled.
41. Screenshot igre, ulaz u selo poslije mosta.
42. Screenshot igre, izlaz iz sela, srušen most.
43. Screenshot igre, kraj levela demo-a, ali ne cijele igre.
44. Screenshot igre, kuća kovača.
45. Screenshot igre, kuća klesara.
46. Screenshot igre, dolazak nazad na početak levela.
47. Screenshot igre, izgled kraja igre.
48. Screenshot igre, izgled menia pauze.

Programi

Photoshop 2015, UnrealEngine 5, BoscaCoil.

Izvori

Photoshop Dokumentacija

Unreal Engine Dokumentacija

<https://www.techartget.com/whatis/definition/role-playing-game-RPG>

<https://en.wikipedia.org/wiki>

https://hr.wikipedia.org/wiki/The_Witcher

<https://www.metacritic.com/game/pc/shaolin-vs-wutang-2/details>

https://bindingofisaac.fandom.com/wiki/The_Binding_of_Isaac_Wiki

<https://docs.unrealengine.com/5.2/en-US>

Prilozi

Slike unutar dokumenta

Zic_Boris_Zavrzni.docx/pdf

Zic_Boris_PPZ.pptx

Složeni demo igre